An Alternate Arithmetic for Word Vector Analogies

Silviu Pitis
silviu.pitis@gmail.com

June 7, 2016

Abstract

In recent years, word vector arithmetic of the type `king - man + woman` $\approx$ `queen` has been applied to solve an analogy task in order to evaluate word embeddings produced by various algorithms. This paper investigates the idea of treating the relationships between word vectors as rotations of the embedding space instead of as vector differences, and shows that such treatment can produce better average approximations of target words.

1 Introduction

In recent years, word vector arithmetic of the type `king - man + woman` $\approx$ `queen` has been applied to an analogy task in order to evaluate word embeddings produced by various algorithms (Mikolov et al., 2013b; Mnih and Kavukcuoglu, 2013; Pennington et al., 2014). Embeddings that do well on this task are said to capture *linguistic regularities* (Mikolov et. al) or *relational similarities* (Turney, 2006).

Such arithmetic is mysterious in the following sense: before taking the vector sum, `king - man + woman`, the literature normalizes each vector so that it lies on the unit hypersphere (i.e., $\|\boldsymbol{x}\| = 1$), but generally, the vector sum `king - man + woman` does not itself lie on the unit hypersphere. Although the similarity between vectors is measured by cosine distance, which ignores lengths, it seems odd that the best way to combine `king`, `queen` and `woman` is to produce a vector not of length 1.

To illustrate the two dimensional case, consider *Figure 1*, where we see that the vector sum `king - man + woman` ends up inside the unit circle. We would need to normalize the sum to obtain an approximation for the `queen` unit vector, but this can alter relationships (note how `queen - woman` and `king - man` do not point in the same direction).

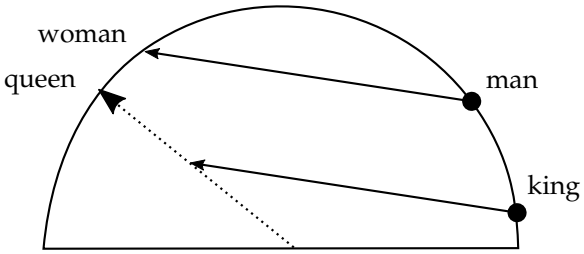

Figure 1: $\|$`king - man + woman`$\| < 1$.

By treating the relationship between `man` and `woman` as a rotation of the embedding space instead of a difference, I show that we can apply that same rotation to the vector for `king` to achieve, on average, a better vector approximation for `queen`.

Using pretrained word embeddings, generated by word2vec, I show that applying a rotation to the analogy task produces approximations that are closer (as determined by cosine similarity) to the target word in more than 75% of the analogies in the GOOGLE and MSR datasets. In 4 of the 14 subcategories of the GOOGLE dataset, using a rotation beats arithmetic in more than 90% of cases.

Although the *approximations* produced by applying a rotation are better on average, the *predictions* generated by selecting the closest words from the vocabulary to such approximation are mixed. When considering the single closest word, simple arithmetic beats rotations in most cases. When considering the ten closest words, however, the rotation approach is superior.

2 Problem Description

Let $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$ be the word embedding vectors (all normalized to unit norm)[1] in the analogy, $\boldsymbol{a}$ is to $\boldsymbol{b}$ as $\boldsymbol{c}$ is to $\boldsymbol{d}$. The analogy task asks us to pick the best $\boldsymbol{d}$ given $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$. For example, $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$ might be the word embeddings for `man`, `woman`, `king`, `queen`, respectively.

Mikolov et al. (2013a) show that embeddings trained by word2vec achieve surprisingly accurate results on this task through the simple arithmetic and cosine similarity.

That is, they choose $\boldsymbol{d}$ as:

$$\boldsymbol{d} = \arg\max_{\boldsymbol{d}^* \in \mathcal{V}} \left(sim(\boldsymbol{d}^*, \boldsymbol{c} - \boldsymbol{a} + \boldsymbol{b}) \right)$$

where

$$sim(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{\|\boldsymbol{u}\| \|\boldsymbol{v}\|}.$$

This is odd because $\boldsymbol{c} - \boldsymbol{a} + \boldsymbol{b}$ is serving as a prediction for $\boldsymbol{d}$, but $\|\boldsymbol{d}\| = 1$ and, in general, $\|\boldsymbol{c} - \boldsymbol{a} + \boldsymbol{b}\| \neq 1$. We could normalize the prediction, but as can be seen in *Figure 1*, normalizing the prediction alters the direction of linear differences such as `queen` $-$ `woman`. *Figure 2* shows the 3-dimensional case.

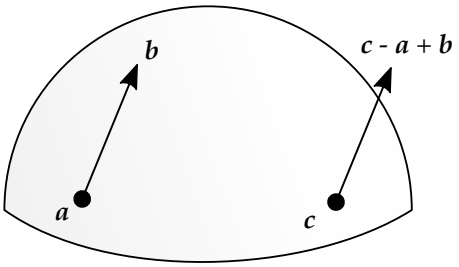

Figure 2: Simple arithmetic in 3 dimensions.

We can overcome this problem by treating the relationship between word vectors as a norm-preserving rotation of the d-dimensional embedding space, $\boldsymbol{R} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ rather than as a linear difference, because $\|\boldsymbol{R}\boldsymbol{x}\| = \|\boldsymbol{x}\| \; \forall \boldsymbol{x} \in \mathbb{R}^d$.

By treating the relationships between word vectors in this way, I show that we can, on average, improve the approximation for $\boldsymbol{d}$, as measured by cosine similarity.

3 Models

In this section, I define the 3 candidate rotations to be applied to $\boldsymbol{c}$ in order to approximate $\boldsymbol{d}$.

3.1 Rotation from a to b

The analogy, $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}$, says that the relationship between $\boldsymbol{a}$ and $\boldsymbol{b}$ is the same as the relationship between $\boldsymbol{c}$ and $\boldsymbol{d}$. Thus, if $\boldsymbol{b}$ is a function, f, of $\boldsymbol{a}$, $\boldsymbol{b} = f(\boldsymbol{a})$, then by hypothesis, $\boldsymbol{d} = f(\boldsymbol{c})$.

Define $\boldsymbol{R}_{a \rightarrow b}$ as the rotation of the embedding space using the plane containing the origin, $\boldsymbol{a}$ and $\boldsymbol{b}$ as the plane of rotation that maps $\boldsymbol{a}$ onto $\boldsymbol{b}$. It will be a rotation by θ, where $\cos\theta = sim(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a} \cdot \boldsymbol{b}$.

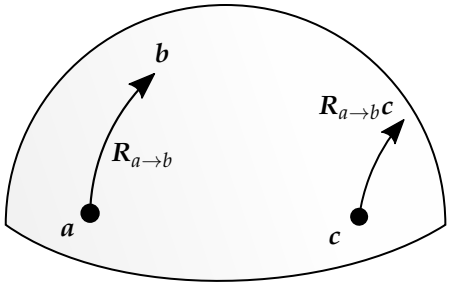

Figure 3: $\boldsymbol{R}_{a \rightarrow b}$ in 3 dimensions.

Then $\boldsymbol{b} = \boldsymbol{R}_{a \rightarrow b}\boldsymbol{a}$, and we can *approximate* $\boldsymbol{d}$ as $\boldsymbol{R}_{a \rightarrow b}\boldsymbol{c}$ and *predict* $\boldsymbol{d}$ as:

$$\boldsymbol{d} = \arg\max_{\boldsymbol{d}^* \in \mathcal{V}} \left(sim(\boldsymbol{d}^*, \boldsymbol{R}_{a \rightarrow b}\boldsymbol{c}) \right).$$

This is illustrated in *Figure 3*.

3.2 Midpoint rotation

If $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}$ then $\boldsymbol{c} : \boldsymbol{d} :: \boldsymbol{a} : \boldsymbol{b}$. But $\boldsymbol{b} = \boldsymbol{R}_{a \rightarrow b}\boldsymbol{a}$ whereas $\boldsymbol{d}$ is only approximated by $\boldsymbol{R}_{a \rightarrow b}\boldsymbol{c}$.

Instead, consider the rotation of the embedding space, $\boldsymbol{R}_{mid}$, that maps the normalized midpoint of the segment ($\boldsymbol{a}$,$\boldsymbol{c}$) onto the normalized midpoint of the segment ($\boldsymbol{b}$, $\boldsymbol{d}$). For such $\boldsymbol{R}_{mid}$, we expect that:

$$sim(\boldsymbol{b}, \boldsymbol{R}_{mid}\boldsymbol{a}) = sim(\boldsymbol{d}, \boldsymbol{R}_{mid}\boldsymbol{c}).$$

.

This is illustrated in *Figure 4*.

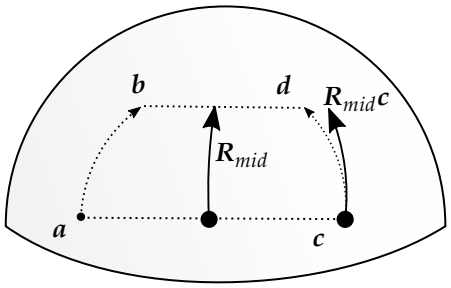

Figure 4: $\boldsymbol{R}_{mid}$ in 3 dimensions.

[1] The literature generally normalizes the vectors before adding them. See Mikolov et al. (2013b) and Mnih and Kavukcuoglu (2013). The author's own experiments confirmed that using unit vectors produces better results on the analogy task described.

For the analogy task we don't know $\boldsymbol{d}$ and cannot precisely calculate $\boldsymbol{R}_{mid}$. However, we can approximate it by substituting an approximation for $\boldsymbol{d}$, such as $\boldsymbol{R}_{a\to b}\boldsymbol{c}$.

Therefore, we choose:

$$\boldsymbol{R}_{mid} = \boldsymbol{R}_{p\to q}, \text{ where}$$
$$\boldsymbol{p} = \frac{a+c}{2}, \text{ and } \boldsymbol{q} = \frac{b+\boldsymbol{R}_{a\to b}\boldsymbol{c}}{2}$$

as the rotation of the embedding space using the plane containing the origin, $\boldsymbol{p}$ and $\boldsymbol{q}$ as the plane of rotation that maps $\frac{\boldsymbol{p}}{\|\boldsymbol{p}\|}$ onto $\frac{\boldsymbol{q}}{\|\boldsymbol{q}\|}$.

3.3 Three point rotation

A third candidate asserts that if $\boldsymbol{a} : \boldsymbol{b} :: \boldsymbol{c} : \boldsymbol{d}$ then $\boldsymbol{a} : \boldsymbol{c} :: \boldsymbol{b} : \boldsymbol{d}$. The latter analogy is often nonsense (or rather arbitrary), so we wouldn't expect this to work that well, except in special cases. In any case, if this were true, then the following transformation should produce $\boldsymbol{d}$ when applied to $\boldsymbol{c}$:

$$\boldsymbol{R}_{3point} = \boldsymbol{R}_{c\to a}^{-1}\boldsymbol{R}_{a\to b}\boldsymbol{R}_{c\to a}$$

This is illustrated in *Figure 5.*

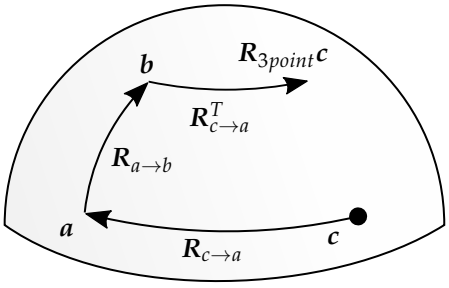

Figure 5: $\boldsymbol{R}_{3point}$ in 3 dimensions.

4 Experiments

4.1 Experiment 1: Analogy Closeness

For $R_i \in \{\boldsymbol{R}_{a\to b}, \boldsymbol{R}_{mid}, \boldsymbol{R}_{3point}\}$, the first experiment compares:

$$sim(\boldsymbol{d}, \boldsymbol{c} - \boldsymbol{a} + \boldsymbol{b})$$
to
$$sim(\boldsymbol{d}, \boldsymbol{R}_i\boldsymbol{c})$$

where $\boldsymbol{d}$ is the word vector for the true value. This experiment compares the approximation to the true value; it does not produce a prediction by considering all possible values in the vocabulary to pick the closest value, which task is performed in Section **4.3**.

The word vectors used were pre-trained word2vec embeddings made available by Google[2], which use a vocabulary size of 3 million and 300 dimensional embeddings.

The analogies used were taken from the **GOOGLE** dataset of 19544 analogies[3] and the **MSR** dataset of 8000 analogies.[4] Of the 8000 analogies in the full **MSR** dataset, 978 were skipped due to words not being in the vocabulary, leaving 7022 analogies.[5]

The results show that both $\boldsymbol{R}_{a\to b}$ and $\boldsymbol{R}_{mid}$ produce better approximations than simple arithmetic in a majority of cases for both the **GOOGLE** and **MSR** analogies. See *Table 1* for a summary of the results, where "Avg. Diff." is defined as the average of

$$sim(\boldsymbol{d}, \boldsymbol{R}_i\boldsymbol{c}) - sim(\boldsymbol{d}, \boldsymbol{c} - \boldsymbol{a} + \boldsymbol{b})$$

over all analogies, and "Win %" is the defined as the percentage of analogies for which

$$sim(\boldsymbol{d}, \boldsymbol{R}_i\boldsymbol{c}) > sim(\boldsymbol{d}, \boldsymbol{c} - \boldsymbol{a} + \boldsymbol{b}).$$

It is interesting that the average cosine distance between the approximation and the true result is less for approximations produced by $\boldsymbol{R}_{a\to b}$ than for those produced by $\boldsymbol{R}_{mid}$, but $\boldsymbol{R}_{mid}$ outperforms simple arithmetic in more cases than does $\boldsymbol{R}_{a\to b}$. A direct comparison of $\boldsymbol{R}_{a\to b}$ to $\boldsymbol{R}_{mid}$, using $\boldsymbol{R}_{a\to b}$ as the baseline, in *Table 2*, shows that $\boldsymbol{R}_{a\to b}$ is a better choice for producing approximations despite $\boldsymbol{R}_{mid}$ outperforming simple arithmetic in more cases.

4.2 Experiment 2: Closeness by Category

The second experiment compares the winners, $\boldsymbol{R}_{a\to b}$ and $\boldsymbol{R}_{mid}$, of the first experiment to the baseline simple arithmetic approach across each of the 14 types of analogies in the **GOOGLE** dataset. The categories and results are shown below in *Table 3*.

In 4 of the 14 subcategories, using the rotation $\boldsymbol{R}_{mid}$ beats arithmetic in more than 90% of cases.

[2] `code.google.com/archive/p/word2vec/`

[3] `word2vec.googlecode.com/svn/trunk/questions-words.txt`

[4] `research.microsoft.com/en-us/projects/rnn`

[5] The excluded analogies were possessives like `daughter : daughter's :: school : school's`, since `daughter's` and `school's` were not in the vocabulary. One could drop the apostrophe and include these as plurals, but plural relationships are already included in the dataset and certain plurals like `lifes` would not make sense.

That there is such variance between different categories is consistent with the past work of Levy and Goldberg (2014), who show that the accuracy of simple arithmetic on the analogy task ranges from 14.55% (for the `currency` category) to 90.51% (for the `capital-common-countries` category).

4.3 Experiment 3: Evaluating Analogies

This experiment evaluates the effectiveness of simple arithmetic, $\boldsymbol{R}_{a\to b}$ and $\boldsymbol{R}_{mid}$ for solving analogies by finding the nearest neighbors to the approximations they produce. The experiment asks whether $\boldsymbol{d}$ is closest word in the vocabulary, as measured by cosine similarity, to the approximation produced by the model (e.g., $\boldsymbol{R}_{a\to b}\boldsymbol{c}$). Separately, I considered whether $\boldsymbol{d}$ was among the *ten* closest words.

Because the closest word to the approximation is often either $\boldsymbol{b}$ or $\boldsymbol{c}$, and consistent with past literature, I removed $\boldsymbol{b}$ and $\boldsymbol{c}$ from the vocabulary when choosing the closest word. See Mnih and Kavukcuoglu (2013). I did not similarly remove $\boldsymbol{b}$ and $\boldsymbol{c}$ when looking at the ten closest words, as they are not problematic in this case.

The results, in *Table 4*, show that $\boldsymbol{R}_{a\to b}$ performs worse than simple arithmetic when considering only the closest word, but does better when considering the ten closest words. $\boldsymbol{R}_{mid}$ was competitive with simple arithmetic when looking only at the closest word, and outperformed simple arithmetic when looking at the ten closest words.

Although the accuracies of all methods improved when given ten guesses, it is interesting that the improvement of $\boldsymbol{R}_{a\to b}$ is the sharpest; allowing extra guesses lifts $\boldsymbol{R}_{a\to b}$ from last place on both datasets to being the winner of the MSR dataset and the most categories in the GOOGLE dataset.

5 A Note on SemEval-2012 Task 2

SemEval-2012 Task 2,[6] appearing in Jurgens et al. (2012), has been used to measure semantic regularities in word vectors (Mikolov et al., 2013b; Levy and Goldberg, 2014). The SEMEVAL dataset covers 79 categories of semantic relationships. For each category, 3 or 4 prototypical word pairs exemplify the relationship, and the task involves ranking approximately 40 target word pairs in that category according to how well they reflect the relationship. For example, the `CLASS-INCLUSION: Taxonomic` category is exemplified by the word pairs, `flower:tulip`, `emotion:rage` and `poem:sonnet`, and the task is to rank 41 target word pairs including `hair:brown` and `pet:dog` by the degree to which they belong in the same category.

If $\boldsymbol{a} : \boldsymbol{b}$ is a prototypical word pair and $\boldsymbol{c} : \boldsymbol{d}$ is a target word pair, past authors have evaluated the relational similarity in two ways:

1. as an analogy, by evaluating $sim(\boldsymbol{d}, \boldsymbol{c} - \boldsymbol{a} + \boldsymbol{b})$, or
2. as a relationship, by evaluating $sim(\boldsymbol{d} - \boldsymbol{c}, \boldsymbol{b} - \boldsymbol{a})$.

Mikolov et al. (2013b) and Levy and Goldberg (2014) found that the latter option produces better results.

Thinking in terms of rotations changes the former option. That is, instead of evaluating $sim(\boldsymbol{d}, \boldsymbol{c} - \boldsymbol{a} + \boldsymbol{b})$, we can evaluate $sim(\boldsymbol{d}, \boldsymbol{Rc})$. This produces the following results:

	Arithmetic	$\boldsymbol{R}_{a\to b}$	$\boldsymbol{R}_{mid}$
Accuracy	40.7%	37.7%	**41.1%**

The second option, comparing the relationships directly, achieves an accuracy of **44.8%**, outperforming all analogical approaches. Thinking in terms of rotations becomes a bit trickier for this option. We can calculate each relationship as $\boldsymbol{R}_{a\to b}$ and $\boldsymbol{R}_{c\to d}$, but there is no best way to determine their similarity.

One approach is to use a generalized concept of an angle that allows an angle to be defined between two arbitrary subspaces and then rank target word pairs by the angle between the planes of rotation of $\boldsymbol{R}_{a\to b}$ and $\boldsymbol{R}_{c\to d}$. This throws out the direction of the rotation, but we can correct this by multiplying by the sign of $sim(\boldsymbol{d} - \boldsymbol{c}, \boldsymbol{b} - \boldsymbol{a})$. Applying the definition of an angle between flats given by Jordan, the algorithm for which is described in Knyazev and Argentati (2002) and implemented in the `krypy`[7] python package, this approach achieves a relatively respectable accuracy of **43.5%**.

The easier, and perhaps more natural, approach to comparing the rotations is precisely the approach of Mikolov et al. (2013b) described above. The difference vectors $\boldsymbol{b} - \boldsymbol{a}$ and $\boldsymbol{d} - \boldsymbol{c}$ lie in the planes of rotation of $\boldsymbol{R}_{a\to b}$ and $\boldsymbol{R}_{c\to d}$, respectively. In addition to the plane of rotation, the difference vectors also contain information about the *direction* and *magnitude* of the rotation.

[6] `https://sites.google.com/site/semeval2012task2/`
[7] `https://github.com/andrenarchy/krypy`

Category	No. of Analogies	$R_{a\to b}$ Avg. Diff.	$R_{a\to b}$ Win %	R_{mid} Avg. Diff.	R_{mid} Win %
capital-common-countries	506	**0.023**	61.7%	0.022	**83.8%**
capital-world	4524	**0.035**	78.0%	0.022	**85.2%**
city-in-state	2467	**0.022**	**67.2%**	-0.003	50.6%
currency	866	-0.000	**50.2%**	-0.002	47.6%
family	506	**0.028**	**80.2%**	0.011	72.5%
gram1-adjective-to-adverb	992	**0.067**	**85.0%**	0.019	76.1%
gram2-opposite	812	**0.050**	**75.6%**	0.005	60.6%
gram3-comparative	1332	-0.007	40.0%	**0.006**	**62.5%**
gram4-superlative	1122	-0.004	45.7%	**0.006**	**59.1%**
gram5-present-participle	1056	**0.067**	92.4%	0.047	**98.6%**
gram6-nationality-adjective	1599	0.007	51.8%	**0.017**	**88.7%**
gram7-past-tense	1560	**0.063**	87.2%	0.044	**94.1%**
gram8-plural	1332	**0.080**	95.4%	0.049	**96.8%**
gram9-plural-verbs	870	**0.051**	75.9%	0.039	**90.1%**

Table 3: Comparison of $R_{a\to b}$ and R_{mid} to simple arithmetic by analogy category in the **GOOGLE** *dataset*

Category	Closest word Arithmetic	Closest word $R_{a\to b}$	Closest word R_{mid}	Closest ten words Arithmetic	Closest ten words $R_{a\to b}$	Closest ten words R_{mid}
GOOGLE dataset	**54.1%**	41.0%	52.5%	60.7%	61.2%	**62.6%**
MSR dataset	39.1%	30.3%	**40.4%**	42.5%	**48.7%**	47.9%
GOOGLE dataset by category						
capital-common-countries	**61.5%**	49.8%	60.9%	67.0%	62.1%	**68.8%**
capital-world	**62.4%**	43.4%	61.5%	68.5%	68.0%	**71.3%**
city-in-state	**41.1%**	27.1%	35.8%	**46.3%**	**46.3%**	43.5%
currency	**26.8%**	4.5%	21.6%	**38.0%**	17.3%	33.9%
family	**75.1%**	62.6%	72.9%	**86.0%**	84.6%	85.4%
gram1-adjective-to-adverb	**18.0%**	13.3%	17.6%	27.2%	**38.5%**	27.8%
gram2-opposite	**31.2%**	20.1%	26.8%	**37.2%**	31.2%	33.7%
gram3-comparative	56.2%	38.8%	**57.7%**	56.7%	**63.5%**	60.1%
gram4-superlative	**36.3%**	15.0%	31.4%	38.2%	35.1%	**42.6%**
gram5-present-participle	62.4%	62.6%	**64.2%**	75.4%	**83.8%**	82.1%
gram6-nationality-adjective	86.1%	82.9%	**86.2%**	92.3%	92.5%	**92.9%**
gram7-past-tense	**49.0%**	37.9%	47.6%	60.6%	**65.9%**	65.4%
gram8-plural	74.5%	78.0%	**79.4%**	77.0%	**85.5%**	83.9%
gram9-plural-verbs	**49.7%**	20.0%	41.8%	59.5%	50.3%	**61.5%**

Table 4: Accuracy on analogy task at precision levels of 1 and 10.

Therefore, the approach taken by Mikolov et al. (2013b) and Levy and Goldberg (2014) to SemEval-2012 Task 2 is consistent with the idea of thinking about relationships between word vectors as rotations of the embedding space.

6 Conclusion

I've introduced a new way to evaluate and think about the relationships between word vectors: as a rotation of the embedding space instead of as a vector difference. So long as we are using cosine similarity to evaluate the similarity between normalized word vectors, this method makes better geometric sense. When used for the word analogy task, it produces a better average approximation of the target word vector.

These results are neither groundbreaking nor immediately useful from a practical perspective. They do, however, offer a new way to think about and visualize the word embedding space, which may be helpful for future research. Avenues for follow-up may include exploring whether applying rotations to the original unnormalized word embedding can produce better unnormalized approximations, and the use of vector approximations to expand a trained vocabulary with new or foreign words.

References

Jurgens, D., Mohammad, S., Turney, P., and Holyoak, K. (2012). Semeval-2012 task 2: Measuring degrees of relational similarity. *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 356–364.

Knyazev, A. V. and Argentati, M. E. (2002). *Principal Angles Between Subspaces in An A-Based Scalar Product: Algorithms and Perturbation Estimates*, volume 23.

Levy, O. and Goldberg, Y. (2014). Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 171–180. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mikolov, T., tau Yih, W., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics.

Mnih, A. and Kavukcuoglu, K. (2013). Learning word embeddings efficiently with noise-contrastive estimation. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 2265–2273. Curran Associates, Inc.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Turney, P. D. (2006). Similarity of Semantic Relations. *Computational Linguistics*, (February):1–39.